

【特許請求の範囲】

【請求項1】 キャッシュメモリを内蔵したCPUとメモリとを含んで構成される動画像伸長装置により実行される動画像伸長方法であり、動画像伸長用ソフトウェアにより駆動される動画像伸長装置により実行される動画像伸長方法であって、隣接する複数のBピクチャを伸長処理する際に、これら隣接するBピクチャの各々について、対応する位置にある定数(K)個のマクロブロックに対する伸長処理を交互に実行することを特徴とした動画像伸長方法。

【請求項2】 請求項1に記載の動画像伸長方法において、前記Kは1であることを特徴とした動画像伸長方法。

【請求項3】 キャッシュメモリを内蔵したCPUとメモリとを含み構成された動画像伸長装置を動作させるためのプログラムを記録した記録媒体であり、隣接する複数のBピクチャを伸長処理する際に、これら各Bピクチャの各々について、対応する位置にある定数(K)個のマクロブロックに対する伸長処理を交互に実行するステップを含むプログラムを記録した記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は圧縮された動画像データを伸長する技術に関し、特にソフトウェアで動画像データの伸長処理を実行する場合に、キャッシュメモリのアクセスを効率的に行うための技術に関する。

【0002】

【従来の技術】 高品質な動画像データを蓄積したり通信したりするための圧縮および伸長方式としてMPEG (Moving Picture coding Experts Group) と呼ばれる国際標準方式が定められている。MPEGの規格に準拠して圧縮された動画像データをリアルタイムに伸長するためには膨大な計算量を必要とするため、従来は特別に設計されたハードウェアが用いられていた。ところが、最近のマイクロプロセッサの高性能化により、通常のマイクロプロセッサにより、圧縮された動画像データの伸長を、ソフトウェアだけで実現することが可能となりつつある。

【0003】 MPEGの圧縮および伸長アルゴリズムについては、例えば、「1991年4月、コミュニケーションズ・オブ・ザ・エーシーエム、第34巻、第4号 (Communications of the ACM, Vol. 34, No. 4, April, 1991)」に詳しく述べられている。MPEGの伸長アルゴリズムはおおまかにいえば、可変長符号の復号処理、逆量子化処理、逆離散コサイン変換処理、および動き補償処理から成る。

【0004】 図3を参照して、MPEG伸長装置を、図3に示すような、CPU301を中心とするシステムと

CPU301上のソフトウェアの組合せで構成する例について説明する。

【0005】 MPEGで圧縮されたビットストリームは外部記憶装置304に蓄えられる。ビットストリームは、CD-ROM等の外部記憶装置304から読み出され、メモリ302に一時的に格納される。メモリ302に格納されたビットストリームはCPU301上で動作するデコーダソフトウェアの働きにより伸長され、その結果として画像データがメモリ302に格納される。メモリ302に格納された画像データはCPU上のソフトウェアにより表示用フォーマットに変換され、表示装置303に転送される。デコーダソフトウェアは、フロッピディスク装置(FD)307等の外部記憶装置に格納されている。

【0006】 図4は、MPEGで圧縮および伸長をおこなう際の、画像フレームの順序について示したものである。MPEGは、1フレームの画像を符号化するためのピクチャタイプとして、Iピクチャ、Pピクチャ、およびBピクチャという3種類のピクチャタイプを有する。Iピクチャは、動き補償処理が施されず、そのフレーム内の情報だけを用いて圧縮されるピクチャタイプである。Pピクチャは、そのフレームよりも過去のフレームの画像情報から、そのフレームを予測するという動き補償処理が施されるピクチャタイプである。Bピクチャは、そのフレームよりも過去のフレームと未来のフレームとの両方の画像情報から、そのフレームの画像信号を双方向予測するという動き補償処理が施されるピクチャタイプである。

【0007】 図4(A)は、MPEGで動画像を圧縮する際の、典型的なピクチャタイプの割り当てを示している。フレーム#1はIピクチャで、他のフレームの画像情報を参照せずに圧縮される。フレーム#2はBピクチャであり、フレーム#1とフレーム#4を用いた双方向の動き補償処理により圧縮される。フレーム#3もBピクチャであり、フレーム#1とフレーム#4を参照して、双方向の動き補償処理により圧縮される。フレーム#4はPピクチャで、フレーム#1を用いた動き補償処理により圧縮される。フレーム#5はBピクチャで、フレーム#4とフレーム#7を用いた双方向の動き補償処理により圧縮される。フレーム#6はBピクチャで、フレーム#4とフレーム#7を用いた双方向の動き補償処理により圧縮される。フレーム#7はPピクチャで、フレーム#4を参照した動き補償処理により圧縮される。点線の矢印はフレーム間の参照関係を表している。

【0008】 図4(B)は、上で述べたピクチャタイプに従って圧縮および伸長を行う場合の、各フレームの処理順序を示している。圧縮処理を実行する前に、入力動画像信号は、図4(A)から図4(B)のように並び替えられる。

10

20

30

40

50

【0009】MPEGでは、まずIピクチャであるフレーム#1が圧縮される。次に、Pピクチャであるフレーム#4が、フレーム#1の局部復号画像を参照して圧縮される。そして、Bピクチャであるフレーム#2が、フレーム#1とフレーム#4の局部復号画像を参照して圧縮される。またBピクチャであるフレーム#3は、フレーム#1とフレーム#4の局部復号画像を参照して圧縮される。Pピクチャであるフレーム#7は、フレーム#4の局部復号画像を参照して圧縮される。続いて、フレーム#5が、フレーム#4とフレーム#7の局部復号画像を参照して圧縮される。フレーム#6は、フレーム#4とフレーム#7の局部復号画像を参照して圧縮される。

【0010】圧縮した画像を記述するMPEGビットストリームには、フレームを圧縮した順番に各フレームの情報が記述されている。すなわち、伸長器に入力されるフレームの順序は、もともとの動画のフレームの順序とは異っている。

【0011】このようにして圧縮されたMPEGビットストリームを伸長する際には、まずフレーム#1が伸長される。そして、伸長されたフレーム#1の画像を参照してフレーム#4（Pピクチャ）が伸長される。伸長されたフレーム#1と伸長されたフレーム#4と2つの画像を参照して、フレーム#2（Bピクチャ）が伸長される。伸長されたフレーム#1と伸長されたフレーム#4との2つの画像を参照して、フレーム#3（Bピクチャ）が伸長される。伸長されたフレーム#4の画像を参照してフレーム#7（Pピクチャ）が伸長される。続いて、伸長されたフレーム#4と伸長されたフレーム#7との2つの画像を参照してフレーム#5が伸長される。伸長されたフレーム#4と伸長されたフレーム#7との2つの画像を参照してフレーム#6が伸長される。図4（C）は、伸長した画像を、図3の表示装置303等に表示するための順序を示している。

【0012】デコーダソフトウェア、即ち、図3のフロッピーディスク装置に格納されたプログラムによる動作を、図5のフローチャートを参照して、説明する。

【0013】1フレームの伸長処理は、1マクロブロックごとに可変長符号復号（502）、逆量子化（503）、逆DCT（504）、動き補償（505）のステップが繰り返され、1フレームの最後のマクロブロックを処理したら1フレームの伸長処理が終了（507）となる。なお、動き補償処理（505）は、Iピクチャに対しては、行われない。また、動き補償処理（505）では、Pピクチャに対しては、片方向動き補償予測が施され、Bピクチャに対しては、双方向動き補償予測が施される。

【0014】1マクロブロック分の可変長符号復号（502）は、メモリ302に蓄えられたビットストリームを読み出し、数ビットずつ可変長符号テーブルを参照す

ることによってハフマン符号の復号を行う。このとき、メモリ上の場所を指すポインタ（POINTER）がビットストリームを読み出した位置を記憶している。このポインタは、ビットストリームが読み出されると常に読み出された量だけ進められて、常に次に読み出すべきビットストリームの場所を指している。ここで用いられる可変長符号テーブルはメモリ307にデコーダソフトウェアとともにあらかじめ格納されており、可変長符号復号処理を実行する際に、CPU内に読み込まれる。ハフマン符号の復号結果はDCT係数バッファと呼ばれるメモリ302上の一領域に格納される。

【0015】逆量子化（503）は、DCT係数バッファに格納されている値と、MPEGビットストリーム内に含まれている量子化パラメータとの積を求める処理である。逆量子化の結果は、前述のDCT係数バッファに書き戻される。

【0016】逆DCT（504）は、DCT係数バッファに格納されている逆量子化された値に2次元逆離散コサイン変換を施す。2次元逆離散コサイン変換の結果は同じDCT係数バッファに書き戻される。

【0017】動き補償（505）は、動画の時間的冗長性を利用したもので、既に伸長されたフレームから画像の一部を切り出してきて、画像を復元する処理である。切り出される既に伸長済みの画像の位置は、動きベクトルと呼ばれるパラメータ（このパラメータもMPEGビットストリームに含まれている）によって決定される。切り出されて復元される画像は、縦16画素、横16画素からなる、マクロブロックと呼ばれる正方形の領域である。切り出された領域がDCT係数バッファに格納されている逆DCTの結果と加算されることで、マクロブロックの伸長が完了する。伸長されたマクロブロックはメモリ302に格納される。

【0018】図3に示すようなシステムでは、通常、CPU301にキャッシュメモリ305が内蔵されている。キャッシュメモリとは、メモリに比べて、容量は小さいが高速にアクセスできるメモリである。キャッシュメモリには、CPU301が最近アクセスした外部メモリ302の内容が蓄えられている。アクセスしてから長時間アクセスのない内容はキャッシュメモリから追い出されてメモリ302に書き戻されるように制御されている。

【0019】すなわち、CPUが一度アクセスした内容は、その後すぐにアクセスする限りは、キャッシュに内容が残っているのでCPUのマシンサイクル以内に再度アクセスができる。これは、キャッシュヒットと呼ばれる。しかし、まだ一度もアクセスしていない場合あるいは一度アクセスしてから長時間アクセスしなかったためにキャッシュメモリに蓄えられていないメモリの内容をアクセスする場合には、CPUのマシンサイクルの数倍から数十倍の時間がかかる。この現象は、キャッシュミ

スと呼ばれる。

【0020】マイクロプロセッサでは、キャッシュミスによりデータのアクセスを待っている間、プロセッサ全体の動作が止まってしまう。したがって、ソフトウェアの処理速度を向上させるためには、キャッシュミスの発生する確率を可能な限り小とし、CPUの能力を有効に使えるようにすることが重要である。

【0021】現時点でのマイクロプロセッサでは内蔵されているキャッシュメモリの容量は最大数十キロバイトである。システムの構成によっては、CPU301の外側に二次キャッシュメモリと呼ばれる、内蔵されているキャッシュメモリよりも大容量のキャッシュメモリを配置する場合もある。二次キャッシュメモリの容量は現時点では最大数百キロバイトである。一方、MPEGで扱われる画像の1フレームの容量は画面サイズによって数十キロバイトから数メガバイトまで考えられる。MPEG2メインプロファイル・メインレベルと呼ばれる、現在最も汎用的に用いられている規格の場合は1フレームが数百キロバイトとなるので、二次キャッシュには最大でも1フレームの画像しか格納できない。

【0022】従来の技術の伸長装置でMPEG2メインプロファイル・メインレベルの伸長を行なうときの、キャッシュメモリのふるまいについて説明する。例えば図4(B)のフレーム#2を伸長して、次にフレーム#3を伸長する一連の動作について説明する。

【0023】フレーム#2の伸長処理の中で動き補償処理では、フレーム#1とフレーム#4を参照するために、これから伸長されるフレーム#2の画像と合わせて3フレーム分の画像にアクセスする必要がある。キャッシュメモリには3フレーム分の画像は格納できないため、フレーム#2の伸長処理が終了した段階では、キャッシュメモリにはフレーム#1、フレーム#4、およびフレーム#2のそれぞれ一部だけが格納されている。MPEGの伸長アルゴリズムは、画面の左から右下方向に向かって処理を進めていくため、多くの場合、参照されるフレームの伸長済画像も、画面上の上端から下端に向けて順にアクセスされる。したがって、伸長の処理が進むにつれて画面上端に近い部分のデータがキャッシュメモリから追い出されていき、最終的には下端に近い部分のデータだけがキャッシュメモリに残ることになる。

【0024】引き続いてフレーム#3を伸長する処理の動き補償処理において、再度フレーム#1とフレーム#4とを参照する。フレーム#3の画面上端に近い部分を伸長しているときには、フレーム#1とフレーム#4の画面上端に近い部分を参照することになるが、フレーム#2の伸長処理の結果、キャッシュメモリにはフレーム#1とフレーム#4の下端に近い部分のデータしか残っていないため、全てのアクセスがキャッシュミスを引き起こす。また、上端に近い部分の伸長処理を行っている段階で、キャッシュメモリに残っていたフレーム#1と

フレーム#4の下端に近い部分のデータが追い出されるため、フレーム#3の画面下端に近い部分を伸長する処理でも全てのアクセスがキャッシュミスを引き起こすことになる。

【0025】このように従来の技術の動画像伸長装置では、動き補償処理において参照するフレーム全体をキャッシュメモリに格納しておくことができないため、キャッシュミスが多発する。特に隣接する複数のフレームがBピクチャである場合、同一の参照フレームを何度もキャッシュミスを起こしながら読み込まなければならないため、無駄が多い。そのためこの従来技術ではキャッシュミスが多発し、デコードソフトウェアの速度低下の要因となっている。

【0026】

【発明が解決しようとする課題】このように、ソフトウェアで構築された従来の動画像伸長装置では、特に、Bピクチャの伸長処理において、CPU内のキャッシュメモリを有効に使うことができないため、処理速度が遅い。メモリアクセスのための待ち時間がデコードソフトウェアの速度低下の大きな要因となっている。本発明は、Bピクチャの動き補償処理におけるキャッシュミスを抑制し、デコードソフトウェアの速度低下を防止した動画像伸長技術を提供することを目的とする。

【0027】

【課題を解決するための手段】本発明の第1の動画像伸長装置は、あるフレームの伸長処理を行うときに、それがBピクチャでかつ次のフレームもBピクチャならば、1番目のフレームの1マクロブロック分の伸長処理と2番目のフレームの1マクロブロック分の伸長処理を交互に繰り返すことで、隣接する2フレームの伸長処理を同時に行う。本発明の第2の動画像伸長装置は、あるフレームの伸長処理を行うときに、それがBピクチャでかつ次のフレームもBピクチャならば、1番目のフレームのある一定数マクロブロック分の伸長処理と2番目のフレームのある一定数マクロブロック分の伸長処理を交互に繰り返すことで、隣接する2フレームの伸長処理を同時に行う。

【0028】本発明では、隣接する2フレームがともにBピクチャである場合、その2フレームの伸長処理をひとつあるいは複数のマクロブロック単位で交互に行うため、隣接するフレームの同じ位置にあるマクロブロックの伸長処理が連続して実行されることになる。これにより、動き補償処理においてキャッシュミスが発生する確率を低くすることができる。これは隣接するフレームの同じ位置にあるマクロブロックにおける動き補償処理は、参照フレーム中の近い位置にある領域を利用することが多いという性質に基づいている。

【0029】

【発明の実施の形態】以下の説明から明らかとなるように、本発明と従来技術とは、外部記憶装置307に格納

10

20

30

40

50

されたプログラムのみが異なっている。MPEGで圧縮されたビットストリームは外部記憶装置304に蓄えられる。ビットストリームは外部記憶装置304から読み出され、メモリ302に一時的に格納される。メモリ302に格納されたビットストリームはCPU301上で動作するデコーダソフトウェアの働きにより伸長され、その結果として画像データがメモリ302に格納される。フロッピディスク307には、以下で、図1または図2のフローチャートを用いて説明するプログラムが格納されている。

【0030】本発明の第1実施形態の動作について、図を参照して説明する。図1は、本発明の第1実施形態の動作を示すフローチャートである。

【0031】1フレームの伸長処理では、まず、そのフレームのピクチャタイプがBピクチャであるかどうかを判断(102)される。

【0032】入力されたフレームのピクチャタイプが、IピクチャあるいはPピクチャならば、従来の技術と同様に、1マクロブロック分の伸長処理(103;図5のステップ502から505の処理と同一)が繰り返され(103, 107)、1フレームの最後のマクロブロックを処理したら1フレームの伸長処理が終了(108)となる。

【0033】ステップ102での判定の結果、ピクチャタイプがBピクチャの場合、そのフレームをフレームXと呼ぶことにする。その場合、現在のビットストリーム読み出し位置を記憶しているポインタ(POINTER)を変数p1に保存しておき(109)、次のフレームの先頭までビットストリームを読み飛ばす(110)。

【0034】この次のフレームをフレームYと呼ぶことにする。そのときのビットストリーム読み出し位置を変数p2に保存しておきp1に保存してあるビットストリーム読みだし位置をポインタに戻す(111)。

【0035】もし、フレームYのピクチャタイプがBピクチャでなければ(112)、フレームXの伸長処理をIピクチャあるいはPピクチャと同様に行う。

【0036】もし、フレームYのピクチャタイプもBピクチャであれば、まず、フレームXの1マクロブロック分の伸長処理(113;図5のステップ502から505の処理と同一)を行う。

【0037】次に、ポインタを変数p1に保存し、p2に保存されているフレームYの先頭を指すポインタをポインタに戻し(117)、フレームYの1マクロブロック分の伸長処理(118;図5のステップ502から505の処理と同一)を行う。

【0038】次に、ポインタを変数p2に保存し、p1に保存されているビットストリーム読みだし位置をポインタに戻し(122)、フレームXの1マクロブロック分の伸長処理(113;図5のステップ502から50

5の処理と同一)を行う。

【0039】以下同様に、フレームXとフレームYの伸長処理を1マクロブロックごとに交互に繰り返し、最後のマクロブロックを処理したらフレームXとフレームYの2フレーム分の伸長処理が終了(124)となる。

【0040】次に、本発明の第2実施形態の動作について、図を参照して説明する。

【0041】図2は、本発明の第2実施形態の動作を示すフローチャートである。Kをある定数とする。1フレームの伸長処理では、まず、そのフレームのピクチャタイプがBピクチャであるかどうかを判断(202)する。そのフレームのピクチャタイプが、IピクチャあるいはPピクチャならば、従来の技術と同様に1マクロブロック分の伸長処理(203;図5のステップ502から505の処理と同一)が繰り返され(203, 207)、1フレームの最後のマクロブロックを処理したら1フレームの伸長処理が終了(208)となる。

【0042】ステップ202での判定の結果、ピクチャタイプがBピクチャの場合、そのフレームをフレームXと呼ぶことにする。その場合、現在のビットストリーム読み出し位置を記憶しているポインタポインタを変数p1に保存しておき(209)、次のフレームの先頭までビットストリームを読み飛ばす(210)。

【0043】この次のフレームをフレームYと呼ぶことにする。そのときのビットストリーム読み出し位置を変数p2に保存しておき、p1に保存してあるビットストリーム読み出し位置をポインタに戻す(211)。

【0044】もし、フレームYのピクチャタイプがBピクチャでなければ(212)、フレームXの伸長処理をIピクチャあるいはPピクチャと同様に行う。

【0045】もし、フレームYのピクチャタイプもBピクチャであれば、まず、フレームXのマクロブロックK個分の伸長処理(213, 214, 218, 219)を行う。次に、ポインタを変数p1に保存し、p2に保存されているフレームYの先頭を指すポインタを、ポインタに戻し(220)、フレームYのマクロブロックK個分の伸長処理(221, 222, 226, 227)を行う。

【0046】次に、ポインタを変数p2に保存し、p1に保存されているビットストリーム読み出し位置をポインタに戻し(228)、フレームXの1マクロブロック分の伸長処理(213, 214, 218, 219)を行う。

【0047】以下同様に、フレームXとフレームYの伸長処理をK個のマクロブロックごとに交互に繰り返し、最後のマクロブロックを処理したら、フレームXとフレームYの2フレーム分の伸長処理が終了(230)となる。

【0048】ここで、Kはキャッシュメモリのサイズと、伸長する画像のサイズによって最適な値が異なる

10

20

30

40

50

例えば256キロバイトのキャッシュメモリを含む動画伸長装置で横720画素、縦480画素の、MPEG2メインプロファイル・メインレベルの画像を伸長する場合、Kを画面上の横1列のマクロブロックの個数である45とする。

【0049】なお、ここでは隣接する2フレームがともにBピクチャであった場合にその2フレームの伸長処理を交互に行う例について述べたが、3フレーム以上の隣接するフレームが全てBピクチャであった場合に、その3フレーム以上の伸長処理を交互に行うことも可能である。

【0050】

【発明の効果】本発明の効果は、Bピクチャの動き補償処理におけるキャッシュミスが減らすことにより、デコードソフトウェアの速度低下を防ぐことである。本発明の実施の形態では、隣接する2フレームがともにBピクチャである場合、隣接するフレームの同じ位置にあるひとつあるいは複数のマクロブロックの伸長処理が連続して実行されることになる。これにより、動き補償処理においてキャッシュに読み込まれる参照フレームのデータ20を有効利用することが可能となり、キャッシュミスが発*

*生ずる確率を低くすることができる。

【図面の簡単な説明】

【図1】本発明の第1の実施の形態の動作を示すフローチャートである。

【図2】本発明の第2の実施の形態の動作を示すフローチャートである。

【図3】本発明の実施の形態の構成を示すブロック図である。

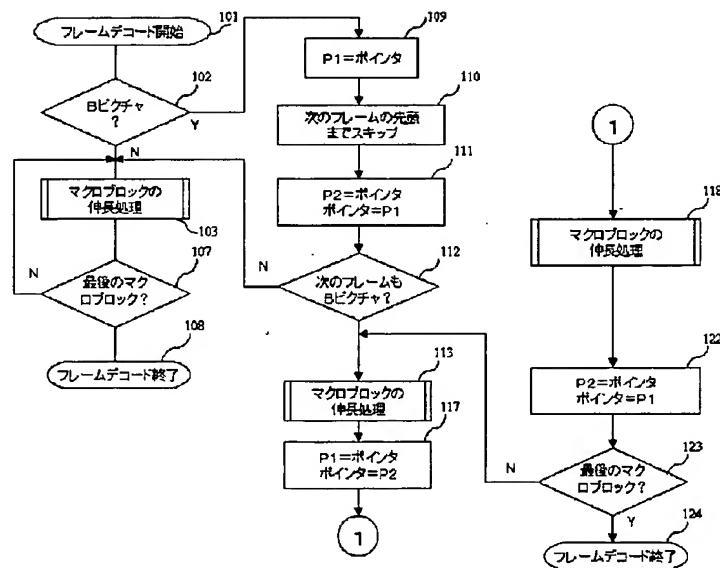
【図4】動画伸長処理におけるフレームの処理順序を示す図である。

【図5】従来の技術の動作を示すフローチャートである。

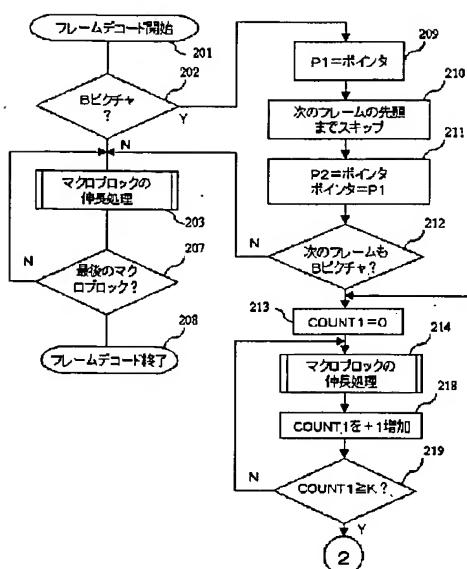
【符号の説明】

301 CPU
302 メモリ
303 表示装置
304 外部記憶装置
305 キャッシュメモリ
306 バス
307 フロッピーディスク装置

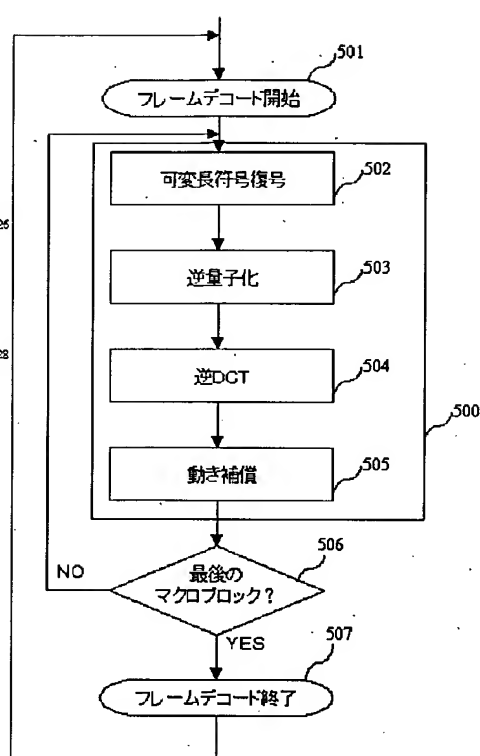
【図1】



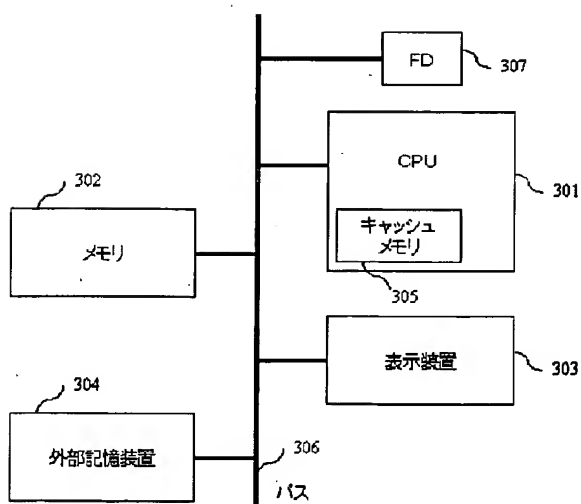
【図2】



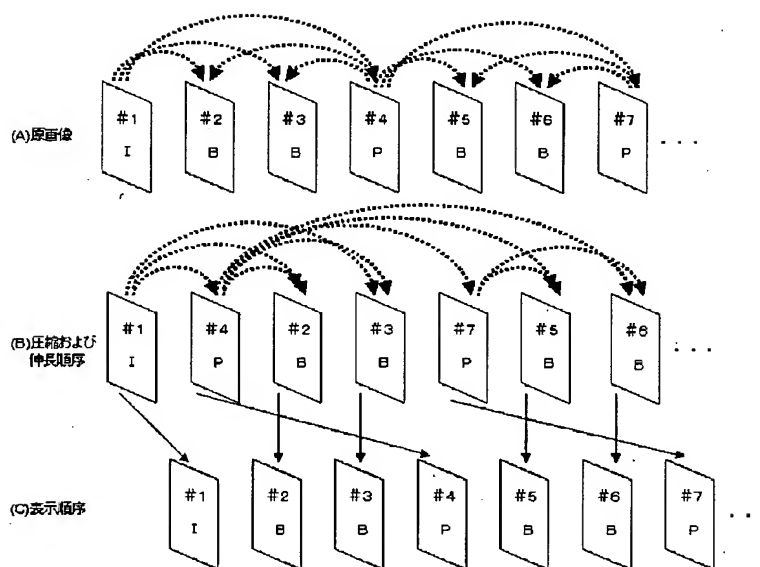
【図5】



【図3】



【図4】



PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-278693

(43)Date of publication of application : 06.10.2000

(51)Int.Cl. H04N 7/32
G06F 12/08

(21)Application number : 11-076130 (71)Applicant : NEC CORP

(22)Date of filing : 19.03.1999 (72)Inventor : IKEGAWA MASAO

(54) MOVING PICTURE EXPANDING METHOD AND RECORDING MEDIUM
RECORDING PROGRAM EXECUTING THE SAME

(57)Abstract:

PROBLEM TO BE SOLVED: To suppress cache errors in the movement compensation of a B picture by alternately repeating expansion of one macro block of a first frame and that of a second frame and simultaneously executing expansion of the two frames when the two adjacent frames are the B pictures.

SOLUTION: When two adjacent frames are B pictures, expansion of one macro block of the first frame and that of the second frame are alternately repeated and the expansion processings of the two frames are simultaneously executed. In the moving picture expansion device, a bit stream compressed by MPEG is read from an outer storage device 304 and is temporarily stored in a memory 302. The stored bit stream is expanded by the operation of decoder software run on CPU 301 and picture data is stored in the memory 302 as the result. A program for expansion processing is stored on

floppy disk 307.

LEGAL STATUS [Date of request for examination] 02.03.2000

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3322233

[Date of registration] 28.06.2002

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

* NOTICES *

JPO and INPIT are not responsible for any

damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.*** shows the word which can not be translated.

3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] It is the dynamic-image expanding approach performed by the dynamic-image expanding equipment constituted including CPU and memory which built in cache memory. It is the dynamic-image expanding approach performed by the dynamic-image expanding equipment driven with the software for dynamic-image expanding. adjoining -- plurality -- B -- a picture -- expanding -- processing -- carrying out -- the time -- these -- adjoining -- B -- a picture -- each -- ***** -- corresponding -- a location -- it is -- a constant -- (-- K --) -- an individual -- a macro -- a block -- receiving -- expanding -- processing -- alternation -- performing -- things -- the description -- ** -- having carried out -- a dynamic image -- expanding -- an approach .

[Claim 2] It is the dynamic-image expanding approach characterized by said K being 1 in the dynamic-image expanding approach according to claim 1.

[Claim 3] cache memory -- having built -- CPU -- memory -- containing --

constituting -- having had -- a dynamic image -- expanding -- equipment --
operating -- making -- a sake -- a program -- having recorded -- a record medium
-- it is -- adjoining -- plurality -- B -- a picture -- expanding -- processing --
carrying out -- the time -- these -- every -- B -- pictures -- each -- ***** --
corresponding -- a location -- it is -- a constant -- (-- K --) -- an individual -- a
macro -- a block -- receiving -- expanding -- processing -- alternation --
performing -- a step -- containing -- a program -- having recorded -- a record
medium .

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to the technique for accessing cache memory efficiently, when performing expanding processing of dynamic-image data especially by software about the technique which elongates the compressed dynamic-image data.

[0002]

[Description of the Prior Art] The international-standards method called MPEG (Moving Picture coding Experts Group) as the compression for storing quality dynamic-image data or communicating and an expanding method is defined. Since computational complexity huge in order to elongate the dynamic-image data compressed based on the specification of MPEG on real time was needed, the hardware designed specially was used conventionally. However, high

performance-ization of the latest microprocessor is enabling it to realize expanding of the dynamic-image data compressed by the usual microprocessor only by software.

[0003] Compression and the expanding algorithm of MPEG are stated to "April, 1991, communique SHONZU OBU THE ESHIEMU, the 34th volume, and No. 4 (Communications of the ACM, Vol.34, No.4, April, 1991)" in detail, for example.

The expanding algorithm of MPEG consists of decode processing of a variable-length sign, reverse quantization processing, reverse discrete cosine transform processing, and motion compensation processing, if it says roughly.

[0004] With reference to drawing 3 , the example constituted from combination of the system centering on CPU301 as shows MPEG expanding equipment to drawing 3 , and the software on CPU301 is explained.

[0005] The bit stream compressed by MPEG is stored in external storage 304. A bit stream is read from the external storage 304, such as CD-ROM, and is temporarily stored in memory 302. The bit stream stored in memory 302 is elongated by the work of decoder software which operates on CPU301, and image data is stored in memory 302 as the result. The image data stored in memory 302 is changed into the format for a display by the software on CPU,

and is transmitted to a display 303. Decoder software is stored in the external storage of floppy disk drive unit (FD) 307 grade.

[0006] Drawing 4 shows the sequence of an image frame at the time of performing compression and expanding by MPEG. MPEG has three kinds of picture types called I picture, P picture, and B picture as a picture type for encoding the image of one frame. I picture is a picture type compressed only using the information in the frame, without performing motion compensation processing. P picture is a picture type with which motion compensation processing in which the frame is predicted is performed from the image information of the past frame rather than the frame. B picture is a picture type with which motion compensation processing in which bidirectional prediction of the picture signal of the frame is carried out is performed from the image information of both the past frame and the frame of the future rather than the frame.

[0007] Drawing 4 (A) shows the assignment of a typical picture type at the time of compressing a dynamic image by MPEG. Frame #1 is I picture and it is compressed, without referring to the image information of other frames. Frame #2 are B picture and they are compressed by the bidirectional motion

compensation processing using frame #1 and frame #4. Frame #3 are B picture and they are compressed by bidirectional motion compensation processing with reference to frame #1 and frame #4. Frame #4 are P picture and they are compressed by the motion compensation processing using frame #1. Frame #5 are B picture and they are compressed by the bidirectional motion compensation processing using frame #4 and frame #7. Frame #6 are B picture and they are compressed by the bidirectional motion compensation processing using frame #4 and frame #7. Frame #7 are P picture and they are compressed by the motion compensation processing which referred to frame #4. The arrow head of a dotted line expresses inter-frame reference relation.

[0008] Drawing 4 (B) shows the processing sequence of each frame in the case of performing compression and expanding according to the picture type described in the top. Before performing compression processing, an input dynamic-image signal is rearranged like drawing 4 (A) to drawing 4 (B).

[0009] In MPEG, frame #1 which is I picture first is compressed. Next, frame #4 which are P picture are compressed with reference to the local decode image of frame #1. And frame #2 which are B picture are compressed with reference to the local decode image of frame #1 and frame #4. Moreover, frame #3 which are

B picture are compressed with reference to the local decode image of frame #1 and frame #4. Frame #7 which are P picture are compressed with reference to the local decode image of frame #4. Then, frame #5 are compressed with reference to the local decode image of frame #4 and frame #7. Frame #6 are compressed with reference to the local decode image of frame #4 and frame #7.

[0010] The information on each frame is described by the sequence which compressed the frame into the MPEG bit stream which describes the compressed image. That is, the sequence of the frame inputted into an expander differs from the sequence of the frame of a dynamic image from the first.

[0011] Thus, in case the compressed MPEG bit stream is elongated, frame #1 is elongated first. And frame #4 (P picture) is elongated with reference to the image of elongated frame #1. Frame #2 (B picture) is elongated with reference to the image of frame #1 elongated, frame #4 which were elongated, and 2 **. Frame #3 (B picture) is elongated with reference to two images of frame #1 elongated and frame #4 which were elongated. Frame #7 (P picture) is elongated with reference to the image of elongated frame #4. Then, frame #5 are elongated with reference to two images of frame #4 elongated and frame #7 which were elongated. Frame #6 are elongated with reference to two images of frame #4

elongated and frame #7 which were elongated. Drawing 4 (C) shows the sequence for displaying the elongated image on the display 303 grade of drawing 3.

[0012] Actuation by the program stored in decoder software, i.e., the floppy disk drive unit of drawing 3, is explained with reference to the flow chart of drawing 5.

[0013] If the step of variable-length sign decode (502), reverse quantization (503), reverse DCT (504), and a motion compensation (505) is repeated for every 1 macro block and expanding processing of one frame processes the macro block of the last of one frame, expanding processing of one frame will be ended (507). In addition, motion compensation processing (505) is not performed to I picture. Moreover, in motion compensation processing (505), to P picture, uni-directional motion compensation prediction is performed and bidirectional motion compensation prediction is performed to B picture.

[0014] The variable-length sign decode (502) for 1 macro block reads the bit stream stored in memory 302, and decodes Huffman coding by referring to a several bits variable-length sign table at a time. At this time, the location where the pointer (POINTER) which points out the location on memory read the bit stream is memorized. When the bit stream was read, only the always read

amount was advanced, and this pointer has pointed out the location of the bit stream which should always be read to a degree. The variable-length sign table used here is beforehand stored in memory 307 with decoder software, and in case it performs variable-length sign decode processing, it is read in CPU. The decode result of Huffman coding is stored in one field on the memory 302 called a DCT multiplier buffer.

[0015] Reverse quantization (503) is processing which asks for the product of the value stored in the DCT multiplier buffer, and the quantization parameter contained in the MPEG bit stream. The result of reverse quantization is returned to the above-mentioned DCT multiplier buffer.

[0016] Reverse DCT (504) gives a two-dimensional reverse discrete cosine transform to the reverse-quantized value which is stored in the DCT multiplier buffer. The result of a two-dimensional reverse discrete cosine transform is returned to the same DCT multiplier buffer.

[0017] A motion compensation (505) is a thing using the time redundancy of a dynamic image, and is processing which cuts down some images from the already elongated frame, and restores an image. The location of the image [finishing / expanding / already] cut down is determined by the parameter (this

parameter is also contained in the MPEG bit stream) called a motion vector. The image cut down and restored is a field of the square called a macro block which consists of 16 pixels long and 16 pixels wide. Expanding of a macro block is completed by being added with the result of reverse DCT by which the started field is stored in the DCT multiplier buffer. The elongated macro block is stored in memory 302.

[0018] In the system as shown in drawing 3 , cache memory 305 is usually built in CPU301. Although capacity is small compared with cache memory in memory, it is the memory which can access a high speed. The contents of the external memory 302 which CPU301 accessed recently are stored in cache memory. After accessing, the contents without long duration access are controlled to ***** from cache memory and to be returned to memory 302.

[0019] That is, since the contents remain in the cache, access of the contents which CPU accessed once is possible, as long as it accesses immediately after that again within the machine cycle of CPU. This is called a cache hit. However, when having not accessed once yet, or in [since long duration access was not carried out once it accessed,] accessing the contents of the memory which is not stored in cache memory, it takes the dozens times [several to] as many time

amount of the machine cycle of CPU as this. This phenomenon is called a cache mistake.

[0020] While waiting for access of data by cache mistake, actuation of the whole processor will stop at a microprocessor. Therefore, in order to raise the processing speed of software, it is important to make into smallness the probability for a cache mistake to occur, as much as possible, and to enable it to use the capacity of CPU effectively.

[0021] In the microprocessor in this time, the capacity of the cache memory built in is 10 K bytes in maximum number. Mass cache memory may be arranged rather than the cache memory which is called secondary cache memory to the outside of CPU301 depending on the structure of a system and which is built in. At present, the capacity of secondary cache memory is 100 K bytes in maximum number. On the other hand, the capacity of one frame of the image treated by MPEG is considered by the screen size from dozens of K bytes to several megabytes. In the case of the specification which is called MPEG 2 Main profile Main level and which is used present the general-purpose, since one frame becomes hundreds of K bytes, only the image of one frame is storable in a secondary cache at the maximum.

[0022] The behavior of cache memory when elongating MPEG 2 Main profile Main level with the expanding equipment of a Prior art is explained. For example, a series of actuation which elongates frame #2 of drawing 4 (B) and then elongates frame #3 is explained.

[0023] In expanding processing of frame #2, in order to refer to frame #1 and frame #4, it is necessary to access the image of frame #2 elongated from now on, and the image for three frames in all by motion compensation processing. the phase which expanding processing of frame #2 ended since the image for three frames was unstorable in cache memory -- cache memory -- each of frame #1, frame #4, and frame #2 -- only the part is stored. In order that the expanding algorithm of MPEG may advance processing caudad toward the right from Hidari of a screen, in many cases, the elongated image of the frame referred to is also accessed in order towards a lower limit from the upper limit on a screen. Therefore, the data of the part near the upper limit of a screen are driven out of cache memory as processing of expanding progresses, and finally only the data of the part near a lower limit will remain in cache memory.

[0024] In motion compensation processing of the processing which elongates frame #3 succeedingly, frame #1 and frame #4 are referred to again. While

elongating the part near the screen upper limit of frame #3, the part near the screen upper limit of frame #1 and frame #4 will be referred to, but since only the data of the part near the lower limit of frame #1 and frame #4 remain in cache memory as a result of expanding processing of frame #2, all accesses cause a cache mistake. Moreover, since the data of the part near [in the phase where expanding processing of the part near upper limit is performed] the lower limit of frame #1 and frame #4 which remained in cache memory are driven out, all accesses will cause a cache mistake also for the processing which elongates the part near the screen lower limit of frame #3.

[0025] Thus, with the dynamic-image expanding equipment of a Prior art, since the whole frame referred to in motion compensation processing is unstorable in cache memory, cache mistakes occur frequently. When two or more frames which adjoin especially are B pictures, in order to have to read a cache mistake for the same reference frame with a lifting repeatedly, there is much futility. Therefore, with this conventional technique, cache mistakes occur frequently and it has become the factor of a rate fall of decoding software.

[0026]

[Problem(s) to be Solved by the Invention] Thus, since cache memory in CPU

cannot be effectively used in expanding processing of B picture especially with the conventional dynamic-image expanding equipment built by software, processing speed is slow. The latency time for memory access is the big factor of a rate fall of decoder software. This invention controls the cache mistake in motion compensation processing of B picture, and aims at offering the dynamic-image expanding technique in which the rate fall of decoding software was prevented.

[0027]

[Means for Solving the Problem] When performing expanding processing of a certain frame, it is B picture, and the 1st dynamic-image expanding equipment of this invention is repeating the expanding processing for 1 macro block of the 1st frame, and the expanding processing for 1 macro block of the 2nd frame by turns, if the following frame's is also B picture, and performs adjoining expanding processing of two frames to coincidence. When performing expanding processing of a certain frame, it is B picture, and the 2nd dynamic-image expanding equipment of this invention is repeating the expanding processing for a fixed number of macroes block with the 1st frame, and the expanding processing for a fixed number of macroes block with the 2nd frame by turns, if

the following frame's is also B picture, and performs adjoining expanding processing of two frames to coincidence.

[0028] In this invention, when both two adjoining frames are B picture, in order to perform the expanding processing of two frames by turns per one or more macro blocks, expanding processing of the macro block in the location where the adjoining frame is the same will be performed continuously. Thereby, the probability for a cache mistake to occur in motion compensation processing can be made low. The motion compensation processing in the macro block which this has in the location where the adjoining frame is the same is based on the property to use the field in the near location in a reference frame in many cases.

[0029]

[Embodiment of the Invention] Only the programs in which this invention and the conventional technique were stored in external storage 307 differ so that it may become clear from the following explanation. The bit stream compressed by MPEG is stored in external storage 304. A bit stream is read from external storage 304, and is temporarily stored in memory 302. The bit stream stored in memory 302 is elongated by the work of decoder software which operates on CPU301, and image data is stored in memory 302 as the result. The program

explained below using the flow chart of drawing 1 or drawing 2 is stored in the floppy disk 307.

[0030] Actuation of the 1st operation gestalt of this invention is explained with reference to drawing. Drawing 1 is a flow chart which shows actuation of the 1st operation gestalt of this invention.

[0031] In expanding processing of one frame, it is judged first whether the picture type of the frame is B picture (102).

[0032] If the inputted picture type of a frame is I picture or P picture, like a Prior art, the expanding processing for 1 macro block (103; the same as that of processing of steps 502-505 of drawing 5) is repeated (103,107), and if the macro block of the last of one frame is processed, expanding processing of one frame will be ended (108).

[0033] The frame will be called Frame X when [in step 102] a picture type is B picture as a result of a judgment. In that case, the pointer (POINTER) which has memorized the current bit stream read-out location is saved at the variable p1 (109), and a bit stream is skipped to the head of the following frame (110).

[0034] This following frame will be called Frame Y. The bit stream readout location which saves the bit stream read-out location at that time at the variable

p2, and has been saved p1 is returned to a pointer (111).

[0035] If the picture type of Frame Y is not B picture (112), expanding processing of Frame X will be performed like I picture or P picture.

[0036] If the picture type of Frame Y is also B picture, expanding processing for 1 macro block of Frame X (113; the same as that of processing of steps 502-505 of drawing 5) will be performed first.

[0037] Next, the pointer which points out the head of the frame Y which saves a pointer at a variable p1 and is saved p2 is returned to a pointer (117), and expanding processing for 1 macro block of Frame Y (118; the same as that of processing of steps 502-505 of drawing 5) is performed.

[0038] Next, the bit stream readout location which saves a pointer at a variable p2 and is saved p1 is returned to a pointer (122), and expanding processing for 1 macro block of Frame X (113; the same as that of processing of steps 502-505 of drawing 5) is performed.

[0039] If expanding processing of Frame X and Frame Y is repeated by turns for every 1 macro block and the last macro block is processed like the following The expanding processing for two frames of Frame X and Frame Y is ended (124).

[0040] Next, actuation of the 2nd operation gestalt of this invention is explained

with reference to drawing.

[0041] Drawing 2 is a flow chart which shows actuation of the 2nd operation gestalt of this invention. Let K be a certain constant. In expanding processing of one frame, it judges first whether the picture type of the frame is B picture (202). If the picture type of the frame is I picture or P picture, the expanding processing for 1 macro block (203; the same as that of processing of steps 502-505 of drawing 5) is repeated like a Prior art (203,207), and if the macro block of the last of one frame is processed, expanding processing of one frame will be ended (208).

[0042] The frame will be called Frame X when [in step 202] a picture type is B picture as a result of a judgment. In that case, the pointer pointer which has memorized the current bit stream read-out location is saved at the variable p1 (209), and a bit stream is skipped to the head of the following frame (210).

[0043] This following frame will be called Frame Y. The bit stream read-out location at that time is saved at the variable p2, and the bit stream read-out location saved p1 is returned to a pointer (211).

[0044] If the picture type of Frame Y is not B picture (212), expanding processing of Frame X will be performed like I picture or P picture.

[0045] If the picture type of Frame Y is also B picture, expanding processing for K macro blocks of Frame X (213, 214, 218, 219) will be performed first. Next, the pointer which points out the head of the frame Y which saves a pointer at a variable p1 and is saved p2 is returned to a pointer (220), and expanding processing for K macro blocks of Frame Y (221, 222, 226, 227) is performed.

[0046] Next, the bit stream read-out location which saves a pointer at a variable p2 and is saved p1 is returned to a pointer (228), and expanding processing for 1 macro block of Frame X (213, 214, 218, 219) is performed.

[0047] If expanding processing of Frame X and Frame Y is repeated by turns for every K macro blocks and the last macro block is processed like the following, the expanding processing for two frames of Frame X and Frame Y will be ended (230).

[0048] Here, the optimal value changes with sizes of the image which K elongates with the size of cache memory. With for example, the dynamic-image expanding equipment containing the cache memory which is 256 K bytes It is referred to as 45 which is the number of a macro block of K of width 1 train on a screen when elongating the image of the MPEG 2 Main profile Main level of 720 pixels wide and 480 pixels long.

[0049] In addition, when both two frames that adjoins here were B picture, the example which performs the expanding processing of two frames by turns was described, but when all of three or more adjoining frames are B pictures, it is also possible to perform the expanding processing of three or more frames by turns.

[0050]

[Effect of the Invention] The effectiveness of this invention is preventing the rate fall of decoding software by reducing the cache mistake in motion compensation processing of B picture. With the gestalt of operation of this invention, when both two adjoining frames are B picture, expanding processing of one or more macro blocks in the location where the adjoining frame is the same will be performed continuously. This is enabled to use effectively the data of a reference frame read into a cache in motion compensation processing, and the probability for a cache mistake to occur can be made low.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is the flow chart which shows actuation of the gestalt of operation of the 1st of this invention.

[Drawing 2] It is the flow chart which shows actuation of the gestalt of operation of the 2nd of this invention.

[Drawing 3] It is the block diagram showing the configuration of the gestalt of operation of this invention.

[Drawing 4] It is drawing showing the processing sequence of the frame in dynamic-image expanding processing.

[Drawing 5] It is the flow chart which shows actuation of a Prior art.

[Description of Notations]

301 CPU

302 Memory

303 Display

304 External Storage

305 Cache Memory

306 Bus

307 Floppy Disk Drive Unit